# DEVELOPMENT OF EPICS DEVICE SUPPORT FOR PIC-BASED I/O CARD WITH A NETWORK INTERFACE

Akito Uchiyama[1,A)], Jun-ichi Odagiri[B)], Misaki Komiyama[C)], Masaki Fujimaki[C)]
[A)] SHI Accelerator Service, Ltd. (SAS)
1-17-6 Osaki, Shinagawa, Tokyo, 141-0032
[B)] High Energy Accelerator Research Organization (KEK)
1-1 Oho, Tsukuba, Ibaraki, 305-0801
[C)] RIKEN Nishina Center for Accelerator-Based Science
2-1 Hirosawa, Wako, Saitama, 351-0198

*Abstract*

In the large experimental facilities such as RIBF, KEKB, and J-PARC, the accelerators are operated by the remote-control system based on EPICS. However, for RIKEN Linear Accelerator (RILAC) of RIBF injector, the control system is not only EPICS-based controllers, but also handmade hard-wired switches in operation. To solve this complication, we developed EPICS device support using asynDriver for PIC Network Interface Card (PICNIC) composed Microchip PIC16F877 and a network interface.

# ネットワーク対応PIC搭載I/Oカード用EPICSデバイスサポートの開発

## １．はじめに

現在の加速器制御システムはRIBF、KEKB、J-PARCといった多くの施設が分散システム構築ツールであるExperimental Physics and Industrial Control System (EPICS)[2]を採用してシステム構築している[1][2][3]。EPICSを用いて構築されたシステムでは上位制御クライアントが各デバイスをEPICS Input/Output Controller (IOC)を介して制御する事により、PLCやGPIB機器、VMEボード等の様々な異なるデバイスが混在しているシステムであっても上位制御側から見ればChannel Accessと呼ばれる同じプロトコルでアクセスする事が可能である。これにより、大型加速器の様な大規模実験施設においてもフロントエンド制御と上位プログラムとの間のインターフェースが統一され保守性、拡張性の高いシステム構築を可能にしている。しかし、IOCからデバイスにアクセスする為には、IOCとデバイスをインターフェースするソフトウェアである、デバイスサポートを用意しなくてはならない。したがってデバイスサポートを自前で開発する手間を省く為に既にデバイスサポートが開発済み、もしくはEPICSコラボレーションでデバイスサポートが用意されているデバイス[3]を選んで採用する事も多い。しかしながらこの事はハードウェアのコスト面に問題がある場合がある。例えばRIBFの入射器であるRILACの電磁石電源、真空系、ビーム診断系の制御システムはほぼEPICSで構築されているが、一部には加速器室とコントロール室を直接配線したハードワイヤのコントローラも存在する[4]。これらはシンプルなON/OFF型スイッチである事が多く、高度なロジックを必要としないが、このスイッチ一つをEPICSでリモート化する為だけにPLCやN-DIM[5]といった高機能なデバイスを採用する事はハードウェアのコストパフォーマンスの面において得策ではない。そこで安価で容易に組込みが可能なI/Oボードであるトライステート社PICNIC Ver.2(図1)に注目し、EPICSデバイスサポートを開発する事とした。

図 1: PICNIC Ver.2 外観

## ２．ターゲットボード

PICNICとはMicrochip社PIC16F877マイコンが内蔵されたイーサネット接続可能なI/Oボードの事である[6]。大きな特長は、PICNICのファームウェアには簡易的なWebサーバが実装されており、別途制御クライアントを開発しなくてもデフォルトでブラウザ

---

[1] E-mail: a-uchi@riken.jp
[2] http://www.aps.anl.gov/epics
[3] http://www.aps.anl.gov/epics/modules/index.php

からアクセスしてI/O制御やPICNIC自身のIPアドレス、ゲートウェイ、ポート番号といった固有設定の変更を行う事が可能、と言う所にある。ネットワーク機器の設定変更が容易にできるユーザフレンドリなインターフェースを持つ事は実際の運用時のメンテナンス性において重要である。

PIC16F877マイコン内に書き込まれるファームウェア、接続ライブラリ、回路図は全てWeb上で公開されており[4]、コントローラ開発の自由度が非常に高く、トライステート社の情報以外にも書籍[5]やWeb上[6]に開発事例の情報が多くあり参考にできる。なお筆者らがPICNICに注目した最大の理由として、安価(2008年7月現在¥7,200-)で購入可能である事に加えて、Linux用ライブラリがGPLで公開されているという点を挙げる。PICNIC Ver.2の仕様[6]を表1に示す。

以上の様な利点のあるPICNICを、EPICSをベースとする制御システムからアクセス可能にする事は有益であると考えられる。

表1：PICNIC Ver.2 仕様

| 通信方式 | Ethernet 10BaseT |
|---|---|
| 内蔵I/F(アナログ) | 入力4ch＋ 温度センサ用1ch<br>入力範囲 0～5 V<br>分解能 10ビット |
| 内蔵I/F(デジタル) | 1.出力4ch, 入力4ch<br>2.出力6ch, 入力2ch (液晶使用時)<br>※1 or 2をPC側で選択可 |
| 内蔵I/F(シリアル) | 入出力1ch |
| 内蔵プロトコル | arp,ip,udp,tcp,http,ICMP,DHCP |

## 3．EPICSデバイスサポート開発手法

### 3.1 フレームワーク

PICNICの様なイーサネットを介してIOCと接続されるデバイスは、VMEやCompact PCIの様なCPUがバスを介してI/Oモジュールにアクセスするデバイスと比べ、I/Oの開始から終了までに時間がかかる為、デバイスサポートを非同期型にしなくてはならない。しかし非同期型のデバイスサポートは通常同期型のそれに比べてプログラムが複雑になり、一から開発する事は負担が大きすぎる。そこでEPICSコラボレーションでは、非同期型のデバイスサポートの実装に共通に必要とされる機能をまとめたフレームワークであるasynDriver[7]が予め用意されている。

一方で、同じような目的でイーサネット接続のデバイスに特化したフレームワークを提供しているnetDev[8]はKEKのEPICS横断化グループと理研加速器制御グループにより開発され、RIBF、KEKB、J-PARCの制御系でも多く利用されている。本開発では既存のPICNIC接続ライブラリを使用する為に低レベルドライバとの切り分けができているasynDriverを選択した。

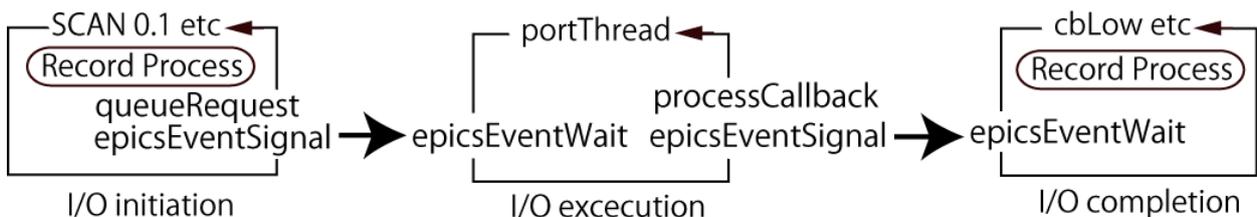
### 3.2 非同期I/Oの仕組み

非同期のデバイスサポートはそれ自体非常に複雑だが、このメソッドはasynDriverの主要部分になるasynManagerによって提供される。非同期I/Oの場合のレコードプロセスの仕組みを具体的に説明すると次の順になる。(図2参照)

・レコードがプロセスされるとデバイスサポートがasynManagerのメソッドの一つであるqueueRequestを呼び、I/O要求をキューに入れI/Oを実行する別スレッド(portThread)にイベントの発生を通知する(epicsEventSignal)
・portThreadはepicsEventWaitでイベント発生を待ち受け、通知された際にI/Oを実行する(processCallback関数の実行)
・processCallback関数の中では非同期I/Oを完成させる為に当該レコードを再度プロセスする為のリクエストを発行する。

上記一連の過程において、デバイスとの通信中に受信待ちでブロックするのはportThreadであり、I/Oを要求したスレッド自身はブロックする事なく、次のレコードプロセスに進める事が非同期I/Oの要点である。

図 2: 非同期I/O の概要

[4] http://www.tristate.ne.jp/picnic/menu.html

[5] 例えば“LANによるハードウェア制御 CQ出版” 等

[6] 例えば“http://wikiwiki.jp/disklessfun/?remote-sys” 等

### 3.3 開発環境

EPICS base, asynDriverのバージョンは現在(2008年2月)の最新を用いて開発を行った。開発環境を含めた詳細を表2に示す。

表2: デバイスサポート 開発環境

| Linux | CentOS 5.1 i386 |
|---|---|
| EPICS base | 3.14.9 |
| asynDriver | R4-9 |
| PICNIC library | libpicnic-20051115.tar.gz |

### 3.4 PICNIC用デバイスサポート

EPICSでPICNICにアクセスする為に開発したデバイスサポートは、DI/DO用のdevBiPicnic, devBoPicnic、AI用のdevAiPicnicである。また、レコードタイプによらない共通の低レベルドライバとしてdrvAsynPicnicを開発した。PICNICとのアクセスはdrvAsynPicnic中でPICNIC接続ライブラリ内の関数を呼んでいる。概要を図3に示す。トライステート社が用意しているLinux版PICNICライブラリでは、I/OピンDI/DO/AIのUDP通信によるアクセスだけでなく、DI/DOピンに接続させる液晶ディスプレイの表示やシリアルポートを利用した通信もサポートされているが、それらの機能はEPICSでは使用頻度が低いと判断した為省いた。よって、本EPICSデバイスサポートで提供される機能はUDPを利用したDI/DO/AIへのI/Oのみである。

| 標準レコード | | |
|---|---|---|
| devBoPicnic | devBiPicnic | devAiPicnic |
| asynManager | | |
| drvAsynPicnic | | |
| PICNIC 接続ライブラリ | | |

図3:PICNIC用デバイスサポート 全体概要
(白抜き部分が本開発で実装された部分)

### 3.5 動作テスト

DI/DO/AIについてテスト用データベースファイルを作成して動作テストを行った。DIテスト用のデータベースファイル"picnic.db"の例を示す。

```
record(bo, "uchiyama:test")
{
        field(DTYP, "Picnic")
        field(OUT, "@picnic1:4")
        field(MASK, "1")
}
```

結果は問題無くCannel AccessによりPICNIC標準搭載のI/O確認用LEDが点灯した。

## 4. まとめ、考察

複雑なロジックが必要ない制御にEPICS対応の安価な組込みI/Oボードを提供する事が可能になり、ハードウェアのコストパフォーマンスの問題を解決できた。asynDriverを用いた開発はデバッグに時間と労力を必要としたが、RILAC制御システムの発展に貢献する事が期待される。なお、今後本デバイスサポートをEPICSコラボレーションにフィードバックしたい、と考えている。

PICNICを遠隔コントローラとして運用していく為にはリレー等回路を組込む必要があり、今後この実装を行う予定である。

## 5. 謝辞

予想を超えて長い開発時間を要する事になってしまったにも関わらず、完成まで継続的なサポートをしてくださった、理研リニアック 大木智則氏、藍原利光氏、山内啓資氏、小山田和幸氏、田村匡史氏に感謝致します。

## 参考文献

[1] M. Komiyama, et al., "CONTROL SYSTEM FOR RIKEN RI-BEAM FACTORY", Proc. 4 Annu. Meet. Particle Accelerator Society of Japan and 32 Linear Accelerator Meet. Japan, Wako, 416-418 (2007).
[2] T. Nakamura, et al., "STATUS OF KEKB ACCELERATORS CONTROL SYSTEM IN 2006", Proc. 3 Annu. Meet. Particle Accelerator Society of Japan and 31 Linear Accelerator Meet. Japan, Sendai, 922-924 (2006).
[3] H. Yoshikawa, et al., "CURRENT STATUS OF THE CONTROL SYSTEM FOR J-PARC ACCELERATOR COMPLEX", Proc. of ICALEPCS07, Knoxville, Tennessee, USA, 2007, 62-64
[4] M. Komiyama, et al., "CONTROL SYSTEM FOR THE RIKEN RI-BEAM FACTORY", Proc. 3 Annu. Meet. Particle Accelerator Society of Japan and 31 Linear Accelerator Meet. Japan, Sendai, 932-934 (2006).
[5] M. Komiyama, et al., "CURRENT STATUS OF THE CONTROL SYSTEM FOR THE RIKEN ACCELERATOR RESEARCH FACILITY", Proc. of ICALEPCS'03, Gyeongju, Korea, 2003, 107-109
[6] トライステート社 PIC Network interface Card Kit Ver.2 取扱説明書
[7] M. R. Kraimer, "EPICS: OPERATING-SYSTEM-INDEPENDENT DEVICE/DRIVER SUPPORT", Proc. of ICALEPCS'03, Gyeongju, Korea, 2003, 205-207
[8] J. Odagiri, et al., "EPICS Device/Driver Support Modules for Network-based Intelligent Controllers", Proc. of ICALEPCS'03, Gyeongju, Korea, 2003, 494-496